

# BEST AVAILABLE COPY

(19)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-228400

(43)公開日 平成10年(1998)8月25日

(51)Int.Cl.  
G 0 6 F 12/00  
G 0 1 C 21/00

識別記号  
5 2 0

F I  
G 0 6 F 12/00  
G 0 1 C 21/00

5 2 0 J  
A

審査請求 未請求 請求項の数43 OL (全 14 頁)

(21)出願番号 特願平9-324429

(22)出願日 平成9年(1997)11月26日

(31)優先権主張番号 0 8 / 7 5 3 4 8 3

(32)優先日 1996年11月26日

(33)優先権主張国 米国 (U.S.)

(71)出願人 000003333

株式会社ゼクセル  
東京都渋谷区渋谷3丁目6番7号

(72)発明者 ジェフ・ジーミン・リヤウ  
アメリカ合衆国・95129・カリフォルニア  
州・サンホゼ・グレン・ヘブン・ドライ  
ブ・1298

(72)発明者 榊原 寿一  
アメリカ合衆国・95014・カリフォルニア  
州・カッパチーノ・ノースショア・スクエ  
ア・10938

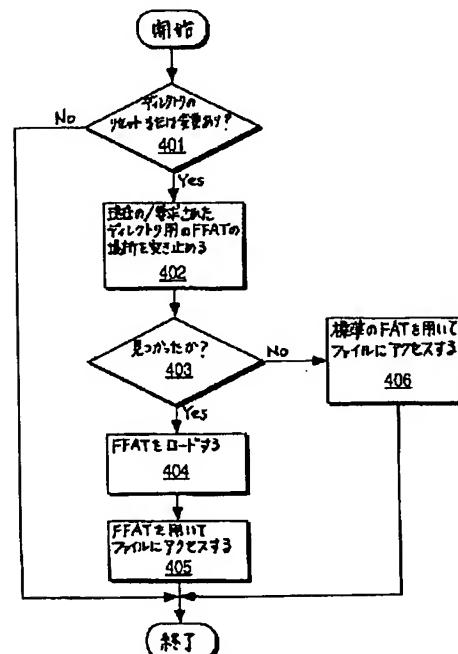
(74)代理人 弁理士 山川 政樹

(54)【発明の名称】 高速ファイルシステム

### (57)【要約】

【課題】 DOSファイルシステムを用いてデータベースのファイルに高速にアクセスする方法を提供する。

【解決手段】 ファイルシステムと結合されたファイルアクセステーブル (FAT) がファイルを参照する。FATは複数の記憶場所を有し、それらの各記憶場所はファイルのデータクラスタの1つ及びFAT中の別の記憶場所に対する参照を含み、それらの参照のリンクされたリストを形成する。ファイル中でデータクラスタが現れる論理順序が、そのファイルについてのFAT中のリンクされたリストをたどることによって決定される。次に、そのファイルに関するクラスタテーブルが構築される。データクラスタに対する参照が、ファイル中のデータクラスタの論理順序に従って、クラスタテーブルの連続した記憶場所に記憶される。次に、クラスタテーブル中の連続した記憶場所に逐次アクセスすることによりファイルにアクセスして、データクラスタが検索され、取り出される。



## 【特許請求の範囲】

【請求項1】 論理順序で配列された複数のサブセットを含むファイルへのアクセスを可能にする方法において：サブセットに対する参照を含み、その参照が前記ファイル中のサブセットの論理順序と異なる順序で配列された第1のテーブルよりファイル中のサブセットの論理順序を決定するステップと；前記ファイル中のサブセットの論理順序に従ってそのサブセットに対する参照を第2のテーブルの連続した記憶場所に記憶するステップと；を具備した方法。

【請求項2】 前記サブセットに対する参照を記憶するステップが：前記第2のテーブル中の各参照の位置が前記ファイル中の対応するサブセットの論理位置を表すようにして、ファイル中のサブセットの論理順序に従ってサブセットに対する参照を第2のテーブルの連続した記憶場所に記憶するステップよりなる；請求項1記載の方法。

【請求項3】 前記第2のテーブル中の複数の連続した記憶場所にアクセスして、前記ファイルのクラスタを検索して取り出すステップをさらに具備した請求項1記載の方法。

【請求項4】 前記第2のテーブル中の記憶場所にランダムにアクセスすることにより前記ファイルにアクセスして、前記ファイルの少なくとも1つのクラスタを検索して取り出すステップをさらに具備した請求項1記載の方法。

【請求項5】 前記ファイルが更新されるのに応答して前記第2のテーブルを自動的に更新するステップをさらに具備した請求項1記載の方法。

【請求項6】 前記参照が、リンクされたリストの形で前記第1のテーブルに記憶され、前記ファイル中のサブセットの論理順序を決定するステップが、そのリンクされたリストで参照を読み取って、サブセットを特定し、かつファイル中のサブセットの論理順序を特定するステップよりなる請求項1記載の方法。

【請求項7】 ファイルシステムを用いてデータベース中の複数のデータサブセットを含むファイルにアクセスできるようにする方法であって、ファイルシステムが複数の記憶場所を有するファイルアクセステーブル（FAT）を含み、各記憶場所がデータサブセットの1つに対する参照及びFAT中の別の記憶場所に対する参照を含んでおり、ファイル中の他の記憶場所に対する参照がファイル中の次の論理データサブセットを指示する方法において：前記FATに基づいて前記ファイル中のデータサブセットの論理順序を特定するステップと；前記論理順序に基づいて前記テーブルの連続した記憶場所に前記クラスタに対する参照を記憶することによってクラスタテーブルを構築するステップと；前記クラスタテーブル中の連続した記憶場所に逐次アクセスすることにより前記データファイルにアクセスして、データファイルのクラスタを検索して取り出すステップと；を具備した方法。

【請求項8】 前記テーブル中の記憶場所にランダムに

アクセスすることにより前記ファイルにアクセスして、ファイルの少なくとも1つのサブセットを検索して取り出すステップをさらに具備した請求項7記載の方法。

【請求項9】 前記テーブルの複数の連続した記憶場所にアクセスして、前記論理順序に従い前記ファイルのクラスタを検索して取り出すステップをさらに具備した請求項7記載の方法。

【請求項10】 前記ファイルの更新に応答して前記テーブルを自動的に更新するステップをさらに具備した請求項7記載の方法。

【請求項11】 FATの記憶場所が、リンクされたリストになっている請求項7記載の方法。

【請求項12】 前記データサブセットが、各々データクラスタである請求項7記載の方法。

【請求項13】 前記データサブセットに対する参照がリンクされたリストとして前記FATに記憶され、前記ファイル中のデータサブセットの論理順序を特定するステップが、リンクされたリストを読み取って、データファイル中のデータサブセットの論理順序を特定するステップよりなる請求項12記載の方法。

【請求項14】 前記ファイルシステムが、DOSファイルシステムである請求項7記載の方法。

【請求項15】 ファイルシステムを用いてデータベース中の複数のデータのクラスタを含むデータファイルにアクセスするための方法であって、ファイルシステムが複数の記憶場所を有するファイルアクセステーブル（FAT）を含み、各記憶場所がデータファイルのクラスタの1つに対する参照及びデータファイルの次のクラスタを指示するFAT中の別の記憶場所に対する参照を含み、データファイルのクラスタに対する参照がFAT中の非連続の記憶場所にあることが可能な方法において：前記データファイル中のクラスタの論理順序を前記FATより特定するステップと；前記論理順序に基づいてクラスタテーブルの連続した記憶場所に前記クラスタに対する参照を記憶することによってクラスタテーブルを構築するステップと；前記クラスタテーブル中の連続した記憶場所に逐次アクセスすることにより前記データファイルにアクセスして、データファイルのクラスタを検索して取り出すステップと；を具備した方法。

【請求項16】 前記データファイルが更新されるのに応答して自動的にクラスタテーブルを更新するステップをさらに具備した請求項15記載の方法。

【請求項17】 前記クラスタに対する参照が、リンクされたリストとしてFATに記憶される請求項15記載の方法。

【請求項18】 前記クラスタの論理順序を特定するステップが、前記リンクされたリストを読み取って、前記データファイル中のデータクラスタの論理順序を特定するステップよりなる請求項17記載の方法。

【請求項19】 DOSファイルシステムを用いてデー

タペース中の複数のデータクラスタを含むファイルにアクセスする方法において：複数の記憶場所を有するDOSファイルアクセステーブル（FAT）にアクセスするステップであって、各記憶場所がデータクラスタの1つ及びFATに対する参照を含み、データクラスタに対する参照がリンクされたリストとしてFATに記憶され、これらの参照が、必ずしも全部はFATの連続した記憶場所に記憶されないステップと；前記リンクされたリストをたどることによってデータクラスタがファイル中に現れる論理順序を特定するするステップと；前記データクラスタに対する参照を含むクラスタテーブルを、前記ファイル中のクラスタの論理順序に従って該クラスタテーブルの連続した記憶場所にデータクラスタに対する参照を記憶することによって、ファイル用のクラスタテーブルを構築するステップと；前記クラスタテーブルの連続した記憶場所にアクセスすることによりファイルにアクセスして、ファイルのクラスタを検索して取り出すステップと；を具備した方法。

【請求項20】 前記データベースが、ある地域の街路の地図データベースである請求項19記載の方法。

【請求項21】 乗物の運転者が進路決定するのを支援するための乗物搭載システムに実装された請求項20記載の方法。

【請求項22】 乗物の使用者が乗物の進路を決定するのを支援するための装置において：プロセッサと；論理順序を有する複数のデータサブセットを含むデータファイルを有するデータベースと；前記プロセッサに接続されたセンササブシステムと；前記プロセッサに接続された前記使用者に進路情報を出力するための出力サブシステムと；を具備し、

前記プロセッサが：前記センササブシステム及び前記データベースからのデータに基づいて乗物の現在位置を計算し；前記出力サブシステムに現在位置の指示情報をユーザに対して出力させる；よう構成されており、前記プロセッサが、さらに：前記データサブセットに対する参照を含み、その参照が前記データファイル中のデータサブセットの論理順序と異なる順序で配列された第1のテーブルよりデータファイル中のデータサブセットの論理順序を決定し；前記データファイル中のデータサブセットの論理順序に従って第2のテーブルの連続した記憶場所に前記データサブセットに対する参照を記憶する；よう構成されている装置。

【請求項23】 前記データベースが、ある地域の街路の地図データベースである請求項22記載の装置。

【請求項24】 前記データファイルが、街路のサブセットを識別する情報を含む請求項23記載の装置。

【請求項25】 前記プロセッサが、さらに、前記第2のテーブル中の各参照の相対的位置が前記論理順序における前記データファイルの対応するクラスタの相対的位置を表すように、データファイル中のデータサブセット

の論理順序に従って第2のテーブルの連続した記憶場所にデータサブセットに対する参照を記憶するよう構成されている請求項22記載の装置。

【請求項26】 プロセッサが、さらに、前記第2のテーブルの記憶場所にランダムにアクセスすることにより前記ファイルにアクセスして、ファイルの少なくとも1つを検索して取り出すよう構成されている請求項22記載の装置。

【請求項27】 前記プロセッサが、さらに、前記第2のテーブルの複数の連続した記憶場所にアクセスすることにより前記ファイルにアクセスして、前記論理順序に従ってファイルのクラスタを検索して取り出すよう構成されている請求項26記載の装置。

【請求項28】 前記プロセッサが、さらに、前記ファイルが更新されるのに応答して自動的に前記第2のテーブルを更新するよう構成されている請求項22記載の装置。

【0029】 前記参照が、リンクされたリストの形で前記第1のテーブルに記憶される請求項2記載の装置。

【請求項30】 プロセッサが、さらに、前記第1のテーブルの参照を読み取ってデータサブセット識別し、かつデータファイル中のデータサブセットの論理順序を特定するよう構成されている請求項記載の装置。

【請求項31】 論理順序に配列された複数のデータサブセットを含むデータベースにおけるデータファイルへのアクセスを可能にする方法ステップを実行するためのコンピュータシステム上で実行可能な命令のプログラムを実装した機械可読プログラム記憶媒体において：参照が、データファイル中のデータサブセットの論理順序と異なる順序で第1のテーブル中に配列されているとき、前記データファイル中のデータサブセットの論理順序をデータサブセットに対するその参照を含む第1のテーブルより決定し、

データファイル中のデータサブセットの論理順序に従つて第2のテーブルの連続した記憶場所に該データサブセットに対する参照を記憶するプログラムを記録した機械可読プログラム記憶媒体。

【請求項32】 前記データサブセットに対する参照を記憶する際に：前記第2のテーブル中の各参照の相対的位置が前記論理順序における前記データファイルの対応するクラスタの相対的位置を表すように該データファイル中のデータサブセットの該論理順序に従つて第2のテーブルの連続した記憶場所にデータサブセットに対する参照を記憶する請求項31記載の機械可読プログラム記憶媒体。

【請求項33】 さらに、前記第2のテーブル中の連続した記憶場所にランダムにアクセスすることにより前記ファイルにアクセスして、前記論理順序に従つてファイルのクラスタを検索して取り出すことを含む請求項31記載の機械可読プログラム記憶媒体。

【請求項34】さらに、前記ファイルが更新されるの応答して自動的に前記第2のテーブルを更新することを含む請求項31記載の機械可読プログラム記憶媒体。

【請求項35】前記参照が、リンクされたリストの形で前記第1のテーブルに記憶される請求項31記載の機械可読プログラム記憶媒体。

【請求項36】前記データファイル中のデータサブセットの論理順序を決定する際に、前記第1のテーブル中の参照を読み取って、データサブセットを特定し、データファイル中のデータサブセットの論理順序を特定する請求項32記載の機械可読プログラム記憶媒体。

【請求項37】プロセッサと；前記プロセッサに接続されたデータベースが記憶されている第1の記憶装置であって、データベースが複数のデータクラスタを含んでいるデータファイルを含み、データファイルが複数の記憶場所を有する第1のテーブルによって参照され、各記憶場所が少なくとも1つのデータクラスタに対する参照及び第1のテーブル中の別の記憶場所に対する参照を含み、データファイルのクラスタに対する参照が、第1のテーブル中の非連続の記憶場所にある第1の記憶装置と；前記プロセッサに接続された第2の記憶装置であって、プロセッサによって実行されるとき、そのプロセッサに：前記データファイル中のデータクラスタの論理順序を前記第1のテーブルより特定させ；前記論理順序に基づいて、その連続した記憶場所に前記クラスタに対する参照を記憶することによって第2のテーブルを構築させ；前記第2のテーブル中の少なくとも1つの連続した記憶場所にランダムにアクセスすることによって前記データファイルにアクセスして、データファイルの少なくとも1つのクラスタを検索して取り出させる；命令を記憶した第2の記憶装置と；を具備したコンピュータシステム。

【請求項38】前記第1のテーブルがDOSファイルシステムによって維持されるファイルアクセステーブル(FAT)である請求項37記載のコンピュータシステム。

【請求項39】前記命令が、さらに、プロセッサによって実行されると、プロセッサに、前記ファイルが更新されるのに応答して自動的に前記クラスタテーブルを更新させる命令を含む請求項37記載のコンピュータシステム。

【請求項40】前記データクラスタに対する参照が、リンクされたリストとして前記第1のテーブルに記憶される請求項37記載のコンピュータシステム。

【請求項41】前記プロセッサに前記データクラスタの論理順序を特定させる前記命令が、さらに、プロセッサに、リンクされたリストを読み取らせることによりデータファイル中のデータクラスタの該論理順序を特定させる請求項37記載のコンピュータシステム。

【請求項42】前記データベースが、ある地域の街路

の地図データベースである請求項37記載のコンピュータシステム。

【請求項43】前記データファイルが、前記街路のサブセットを識別する情報を含む請求項42記載のコンピュータシステム。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、コンピュータシステムに関するものである。より詳しくは、本発明は、搭載型乗物用ナビゲーションシステムのデータベースにアクセスするための高速ファイルシステムに関する。

##### 【0002】

【従来の技術】搭載型ナビゲーションシステムは、自動車の運転者が不案内な土地を通って走る際、進路を決めるのを手助けするのに役立てることができる。このようなシステムの一例が、運転者に記憶された街路地図データベースから目的地を選ばせ、その選択された目的地までのルートを演算させ、その後その選択された目的地へ運転者を導くコンピュータベースの装置である。このようなシステムは、一般に乗物の位置を決定するための種々のセンサを有する。このシステムは、乗物が移動するのに伴って、センサからデータを処理して、地図データに対して最も確からしい乗物の位置を決定する。このシステムは、乗物が移動する際、運転者に周囲の地域の街路地図を表示することができ、また表示された地図上に乗物の位置と動きを示す標識を表示することができる。また、このシステムは、音声及び／または視覚による進路指示を運転者に与えることもできる。このような搭載型自動車用ナビゲーションシステムは、現在、米国カリフォルニア州サンニーベル(Sunnyvale)のゼクセル・ユー・エス・エー・コーポレーション(Zexel USA Corporation)社から入手可能である。

【0003】この種の搭載型ナビゲーションシステムはコンピュータによって制御される。他のコンピュータと同様に、搭載型乗物用ナビゲーションシステムにおけるコンピュータは、システムの基本的な動作を制御するためのソフトウェアであるオペレーティングシステムを有する。オペレーティングシステムは、ファイルに命名し、ファイルを編成し、記憶し、検索する等、ファイルを管理するためのファイルシステムを含んでいる。しかしながら、搭載型乗物用ナビゲーションシステムのファイルシステムに関しては、いくつかの設計上の問題がある。これらの問題は、一般に、ファイルシステムの速度、修正可能性、メモリ使用度、及び市場サポートに関連したものである。

【0004】例えば、相手先ブランド製造業者(OEM)は、外部の供給元からファイルシステムを購入する場合がある。しかしながら、外部ソースから供給される所有権を主張できるソフトウェアの使用は、ファイルシ

システムソフトウェアを変更したり、あるいはデコンパイルする際、供給元によって課されるいくつかの制約の適用を受けなければならないことがある。そのような制約は、動作をデバッグする際の妨げになったり、ナビゲーションシステム全体のアップグレードを妨げたりする。その上、所有権を主張できるファイルシステムについては、市場で支援ツールを見つけることが困難なこともある。のために、オープンアーキテクチャ（すなわち、公表規格）を有するファイルシステムを使用することが望ましい場合がある。

【0005】周知のDOS（ディスクオペレーティングシステム）ファイルシステムは、オープンアーキテクチャを持っており、のためにDOSファイルシステムはこれから搭載型ナビゲーションシステム用ファイルシステムとして魅力的な存在になっている。その上、DOSファイルシステムは、一部の所有権を主張できるファイルシステムと比較して、比較的小さいメモリ量しか必要としない。しかしながら、DOSファイルシステムは、後で詳細で述べるように（「発明の実施の形態」の項に記載）、大きいファイルにアクセスする時、遅くなる傾向がある。ファイルにアクセスするときのこの遅延は運転者の注意をそらし、のために危険を招くことがある。それ故に、標準のDOSファイルシステムは、一般に搭載型乗物用ナビゲーションシステムで使用するのに適切ではない。

【0006】従って、高速データアクセスが可能で、メモリ使用度が小さく、柔軟性及び市場サポートが確保されるファイルシステムが必要とされている。さらに、オープンアーキテクチャを有し、かつ搭載型乗物用ナビゲーションシステムでの使用に適した高速ファイルシステムも必要とされている。

#### 【0007】

【課題を解決するための手段】本発明は、ファイルへのアクセスを可能にする方法である。ファイルは、ファイル中に論理順序に並べられた多数のサブセットを含む。第1のテーブルは、サブセットに対する参照を含んでいる。ファイル中のサブセットの論理順序は、第1のテーブルより決定される。参照は、第1のテーブル中でファイル中のサブセットの論理順序と異なる順序で並べられる。次に、サブセットに対する参照は、ファイル中のサブセットの論理順序に従って第2のテーブルの連続した記憶場所に記憶される。

【0008】本発明のその他の特徴については、添付図面及びこの後の実施形態についての詳細な説明から明らかとなろう。

【0009】以下の、添付図面に示す本発明の実施形態による詳細な説明は、例示説明のためのもので、本発明に対して限定的な意味を有するものではなく、図中、同様の構成要素は同じ参照符号で示してある。

#### 【0010】

【発明の実施の形態】以下、本発明の高速DOSファイルシステムを実施形態により詳細に説明する。以下の説明においては、本発明の完全な理解させるために、具体的な詳細事項を多数記載する。しかしながら、当業者にとっては、本発明がこれらの具体的な詳細事項の記載なしに実施することが可能なことは明白であろう。その他の場合においては、本発明の説明を容易にするために、周知の構成や装置はブロック図形式により示す。

【0011】本発明には、この後詳細に説明するように、DOSファイルシステムの実効データアクセス速度を改善するための技術が含まれる。一実施形態においては、本発明の高速DOSファイルシステムは搭載型自動車用ナビゲーションシステムに実装される。しかしながら、搭載型乗物用ナビゲーションシステムとは無関係なアーキテクチャを含めて、本発明は種々異なるアーキテクチャで実施することができる。例えば、本発明はパーソナルコンピュータに実装することができる。

【0012】全体的なシステムアーキテクチャ  
図1は、本発明を実装した搭載型自動車用ナビゲーションシステム1をブロック図で図解したものである。図示のナビゲーションシステム1は、中央処理装置(CPU)10、リードオンリーメモリ(ROM)11、ランダムアクセスメモリ(RAM)12、及び大容量記憶媒体13を有し、これらはすべてバス24に接続されている。バス24は、実際には、ブリッジ及び/またはバスアダプタによって互いに接続された2本以上のバスとなる場合もある。また、バス24には出力サブシステム26も接続されている。出力サブシステム26は、バス24に接続された音声出力コントローラ15から信号を受け取るがオーディオスピーカ16を含む。また、出力サブシステム26は視覚的表示装置18を含み、この表示装置はバス24に接続されたディスプレイコントローラ17から信号を受信する。バス24には、入力サブシステム14が接続されており、この入力サブシステムは、ユーザ（例えば、乗物の運転者または乗客）が入力をナビゲーションシステム1に入力する手段を与える。また、バス24には、インターフェース装置22を通してセンササブシステム25が接続されている。センササブシステム25は、角速度センサ19、走行マイル数センサ（例えば、走行距離計）20及び全地球側位システム(GPS)の受信器21を有し、これらは各々インターフェース装置22に接続されている。インターフェース装置22は、センサ19、20及びGPS受信器21から受け取ったデータをフォーマットしてバス24へ転送する。

【0013】図1に示す各構成要素については、本発明の範囲内で種々の代替が可能である。例えば、大容量記憶媒体13は、CD-ROM(CD-ROM)(コンパクトディスクROM)、CD-R(記録可能CD)、DVD(デジタル多用途ビデオディスク)等の磁気ディ

スクあるいは光学記憶ディスクを含め、適切なものであれば、任意の形態の不揮発性記憶装置を用いることができる。本発明の一実施形態においては、大容量記憶媒体13は、PCMCIA (Personal Computer Memory Card International Association; パーソナルコンピュータメモリカード国際協会) ハードディスクドライブである。さらに、大容量記憶媒体13、ROM11、及びRAM12は、それぞれ複数の物理装置を表す場合もある。同様に、表示装置18は、ブラウン管(CRT)、液晶表示装置(LCD)、あるいは他の任意の形態の適切な視覚出力装置を用いることができる。一実施形態においては、入力サブシステム14は、ユーザがそれによってアルファニューメリック情報をナビゲーションシステム1に入力することができる、あるいは表示装置18上に表示された情報を選択することができるメカニカルボタンないしはキーを含む。他の実施形態では、メカニカルボタンに代えて感圧ボタンまたは感熱ボタン、タッチセンシティブスクリーン、音声作動入力システム、あるいは別の代替手段を用いることができる。さらに、図1に示す特定のセンササブシステム25は、本発明の範囲内で他の形態の検出装置に置き換えることが可能である。例えば、GPS受信器21は他の形態の高精度測位システムに置き換えることができ、角速度センサに代えて絶対方位検出器(例えば、コンパス)を用いることができる。

#### 【0014】システム全体の動作

ナビゲーションシステム1は、街路地図データ、観光名所、レストラン、空港、ショッピングモールなど閑心対象の場所を識別指示するデータ、及びその他のデータのデータベースを有する。街路地図データは、種々の異なる縮尺及び詳細図の縮尺を表すデータを含む。一実施形態においては、データベースは大容量記憶媒体13に記憶される。このナビゲーションシステム1によって、運転者は、記憶されたデータベースから所望の目的地を選択してから、その選択された目的地へのルートを演算することができる。

【0015】システム1は、乗物の走行中に、表示装置18を用いて所定の地域の街路地図を運転者に対して表示する。この表示には、表示された地図に対する乗物の現在位置と動きを示す指示マークが含まれる。乗物が走行する際、システム1は、角速度センサ19及び走行マイル数センサ20からのデータを処理して、前に演算された位置からの相対位置の変化を求める「推測航法」と呼ばれる技術を利用する。次に、システム1は、地図マッチングアルゴリズムを使って、計算上の位置を地図データベースと比較し、必要に応じて、位置測定を修正する。場合によっては、GPS受信器21からのデータを用いて、推測航法及び地図マッチングを使って得られた測位結果を修正、補正、あるいは置換することもできる。乗物が走行するのに伴って、システムは、ユーザに

対して記録音声または合成音声の形の進路指示をスピーカ16を介して出力する。また、進路指示は表示装置18によりユーザに対して出力することも可能である。

【0016】ナビゲーションシステム1はオペレーティングシステムに基づいて機能する。オペレーティングシステムは、ファイルの命名、編成、記憶、及び検索のような機能を実行するためのファイルシステムを有する。本発明は、以下に説明するように、ファイルシステムの動作を改善するための技術を含むものである。

【0017】一実施形態においては、本発明は、ナビゲーションシステム1でそのCPU10がメモリ(メモリはROM11、RAM12、大容量記憶媒体13、またはこれらの装置の組合せでもよい)に記憶された命令のシーケンスを実行するのに応答して機能する。すなわち、メモリに記憶された命令のシーケンスの実行によって、CPU10は以下に説明する本実施形態のステップを実行する。例えば、命令は、ROM11または大容量記憶媒体13のような永存性記憶装置からRAM12にロードして、実行することも可能である。他の実施形態においては、ソフトウェア命令に代えて、あるいはこれらのソフトウェア命令と組み合わせてハードウェア回路を用いることにより本実施形態を実施することも可能である。このように、本発明は、ハードウェア回路とソフトウェアの何らかの特定の組合せに限定されるものではなく、またナビゲーションシステム1によって実行される命令の如何なる特定のソースに限定されるでもない。

#### 【0018】ファイルアクセステーブル

本発明には、DOSファイルシステムの実効速度を改善するための技術が含まれる。DOSファイルシステムは、そのデータアクセス速度が比較的遅いために、一般に自動車用ナビゲーションシステム用としては適当ではない。ここで、このデータアクセス速度が比較的遅い理由について述べる。

【0019】データは、一般にデータクラスタの形で記憶媒体上に記憶される。クラスタは、記憶媒体のいくつかの連続した物理セクタを意味する。各ファイルのデータクラスタは、そのファイル中において論理順序を有する。DOSファイルシステムは、一般にクラスタチェーンを用いてファイルにアクセスする。クラスタチェーンは、特定のファイルにおけるデータクラスタのリンクされたリストである。これらのクラスタチェーンは、ファイルとそのファイルのデータクラスタの間の結合であるファイルアクセステーブル(FAT)に記憶される。

【0020】図2には、FAT30が図解されている。FAT30は、多数の記憶場所を有し、これらの各記憶場所には、ファイルの特定のデータクラスタに対する参照が記憶される。各参照は、ファイル中の論理順序による次のデータクラスタに対する参照にもなっている。各ファイルにおけるクラスタチェーンの始めは入口点によ

って示され、この入り口点はFAT30における記憶場所12である。従って、File\_Aのクラスタチェーンは(13、14、45、46)である。ファイルのクラスタチェーンの最後のクラスタは、FAT30の記憶場所46に記憶された識別子「LAST」のようなクラスタチェーンの最後の記憶場所に記憶された適切な識別子によって識別される。

【0021】ファイルは、FATでそのファイルと結び付けられたクラスタチェーンを通して逐次順方向検索を行うことによってアクセスされる。現在参照されているデータクラスタを指示するファイルポインタは、クラスタチェーンに沿って前方へ動かすことができる。関連するデータがファイルポインタの後ろにある場合は、ファイルポインタをクラスタチェーンの始めにセットして、探索を再始動することができる。ファイルが小さい場合は、この探索を実行するのに必要な時間は無視できるほど小さいこともある。しかしながら、データベースファイルは一般に大量のデータが入っているので、クラスタチェーンの探索は比較的長い時間を要することが多い。従って、DOSファイルシステムと関連したFATは、短いクラスタチェーンを有する比較的小さいファイルにアクセスするのに適しているかもしれない。しかしながら、FATは、搭載型乗物用ナビゲーションシステムのデータベースに通常見られるような大きいデータファイルにアクセスするのには全く不適である。

【0022】高速ファイルアクセステーブル  
本発明は、(機能上)ファイルシステムのFAT「の上に」置かれた追加構造を生成することによって、効果的にDOSファイルシステムのデータアクセス速度を改善するものである。本願においては、この構造を「高速FAT(FFAT)」と称する。FFATはの一例を図3Aに示す。この構造は、各ファイル毎のクラスタチェーンが、本願で「クラスタテーブル」と称する新しいテーブルに記憶される。クラスタテーブルは、ファイル中の実クラスタに対する参照が書き込まれる多数の記憶場所、すなわち論理クラスタを有する。DOSファイルシステムの標準のFATと対比して、クラスタに対する参照は、ファイル内のクラスタの論理順序に従ってクラスタテーブルの連続した記憶場所に記憶される。その結果、ファイルは、FATを用いた逐次方式ではなく、そのクラスタテーブルを用いてランダム方式によりアクセスすることができる。以下に説明するように、所定のファイルについてのクラスタテーブルは、そのファイルと関連したFATをトラバース、あるいは走査することにより生成される。

【0023】次に、図3Aにおいて、FFATは、記述子34、参照テーブル35、及び「filename\_A」、「filename\_B」、及び「filename\_C」のようないくつかのファイルについてのクラスタテーブル36で構成されている。参照テーブル35

には、各ファイルのファイル名及び各ファイル毎のクラスタテーブルに対する参照が書き込まれる。参照テーブル35中のファイル名は、参照符号「31」によって示し、クラスタテーブルに対する参照は参照符号「32」によって示してある。記述子34は、FFAT中のファイル数及び参照テーブル35のサイズを記述する。このように、FFAT33中の各ファイルは、参照テーブル35にエントリを有する。エントリはファイル名によって識別することができ、そしてファイルのクラスタテーブルは、その関連する参照によって場所を突き止めることができる。このようにして、所定のファイルについてのクラスタテーブルをそのファイル名によってアクセスすることができる。

【0024】図3Bには、クラスタテーブルがより詳細に示されている。クラスタテーブル37は、「論理クラスタ」と呼ばれるN個の記憶場所38を有する。クラスタテーブル37中の各論理クラスタ38は、0、1、2、・・・、N-1の範囲内にあるインデックス値によって識別される。各論理クラスタのインデックス値は、その論理クラスタによって参照されたデータクラスタがデータファイル中に論理順序で現れる順序を表す。例えば、図3Cには、図2によって説明した一例のファイルであるFile\_Aと関連したクラスタテーブル39が示されている。前に述べたように、File\_Aについてのクラスタチェーンは(13、14、45、46)である。このように、ファイルAのクラスタテーブル39は、論理クラスタ0、1、2、と3として識別される(インデックスされる)4つの論理クラスタを含み、これらの論理クラスタにはそれぞれデータクラスタ13、14、45、と46に対する参照が含まれている。

【0025】一実施形態においては、FFATは、利用可能な時だけ、DOSファイルシステムにより使用される任意選択機能として扱われる。FFATが利用可能ではない場合、ファイルシステムは、単に標準のFATを使ってファイルにアクセスする。この実施形態では、システム1がリセットされたたびに、ファイルシステムはデフォルトディレクトリ用のFFATをロードすることを試みる。同様に、ディレクトリの変更が行われたたびに、ファイルシステムは新しいディレクトリ用のFFATテーブルをロードすることを試みる。ディレクトリの変更は、例えば、現在アクセス中の地域と異なる地域を表すデータにアクセスするのに必要になる。FFATのローディングがうまく行われれば、コンフィギュレーションファイルで指定されているファイルについての標準のFATのテーブルアクセスに換えて、FFATのアクセスが行われる。FFATテーブルがうまくロードされないか、ファイルがコンフィギュレーションファイルで指定されていない場合は、通常のFATのアクセスが行われることになる。従って、ナビゲーションシステム1はFATテーブルなしでも動作することができる。ただ

し、その速度はF F A T テーブルを使用する場合より遅くなる。

【0026】次に、この機能について、F F A T の場所を突き止めるためのルーチンを示す図4を参照して説明する。ナビゲーションシステム1のリセットに応答して、あるいはディレクトリの変更と同時に（ステップ401）、ナビゲーションシステム1は、現在のディレクトリまたは要求されているディレクトリ用のF F A T の場所を突き止めようとする（ステップ402）。そのF F A T が見つかると（ステップ403）、システムはその該当するF F A T をロードする（ステップ404）。次に、システムはそのF F A T を用いて適切なファイルにアクセスする（ステップ405）。F F A T が見つからない場合は、システムは標準のF A T を用いて適切なファイルにアクセスする（ステップ406）。

【0027】F F A T によって参照されたデータファイルが更新される毎に、F F A T も更新されなければならない。図5には、F F A T を更新するためのルーチンが示されている。一実施形態においては、図5の手順はこの目的のための専用ソフトウェアユーティリティで実施される。まず、データファイルが入っているディレクトリのパスが入力される（ステップ501）。次に、更新されたデータファイルの名前が、ディレクトリのコンフィギュレーションファイルより検索して取り出される（ステップ502）。次に、更新されたF F A T が、F A T によって指定されたクラスタチェーンを走査することによってF A T から生成される（ステップ503）。F F A T を更新するステップ（ステップ503）は、それによって影響されたファイルのクラスタテーブルを更新する操作を含む。

【0028】図6は、ファイルのクラスタテーブルを生成する、あるいは更新するためのルーチンを示したものである。インデックス変数Iが始めにゼロにセットされ（ステップ601）、システム1は、ファイルポインタに基づいてF A T の第1の記憶場所にアクセスして、第1のクラスタに対する参照を得る（ステップ602）。次に、この参照はクラスタテーブルの論理クラスタLC [1]（Iのインデックス値を有する論理クラスタ）に記憶される。参照がクラスタチェーンの最後のクラスタならば（ステップ604）ルーチンは終了する。最後のクラスタでなければ、インデックス変数Iが1だけインクリメントされる（ステップ605）。インデックス値Iをインクリメントした後、現在の参照からクラスタチェーンにおける次の参照の記憶場所が決定される（ステップ606）。次に、F A T における次の記憶場所がアクセスされて、そのクラスタチェーンにおける次のクラスタに対する参照が得られる（ステップ607）。その後、ステップ603からルーチンが繰り返される。

【0029】データファイルは、ファイルのクラスタテーブルの論理クラスタを逐次読み取ることによってアクセ

スされる。データアクセスは、ファイルシステムへの読み取り要求によって開始される。読み取り要求には、ファイル名、オフセット、及びカウントが含まれる。オフセットは、ファイルの始め、現在位置あるいはファイルの終わりのような、ファイル中の所定の点から測定される。カウントは、読み取られるデータバイト数を指定し、従って、一実施形態においては、読み取り要求は次のようなフォーマットを有する：R E A D ([FILE\_N A M E]，[O F F S E T]，[C O U N T]）。オフセットを与えて、クラスタテーブルの開始論理クラスタがオフセットをクラスタサイズで割ることによって計算される。このように、クラスタテーブルの開始論理クラスタは、逐次方式ではなく、ランダムにアクセスされる。次に、実データクラスタに対する参照をクラスタテーブルの論理クラスタから検索して取り出すことができる。オフセットと関連したセクタ番号は、クラスタテーブルから得られた実クラスタ番号にクラスタ当たりのセクタ数を掛けることによって計算することができる。論理セクタ番号及びセクタ数をデバイスドライバに供給することにより、論理セクタを物理セクタにマップして、データを記憶装置から検索して取り出すことができる。クラスタテーブル中の論理クラスタは、クラスタテーブル中に現れる順に読み取られる。このように、ファイルは、順次アクセスファイルとしてではなく、ランダムアクセスファイルとしてアクセスされる。

【0030】図7は、上に述べた方法でデータ読み取るためのルーチンの例を示したものである。始めに（ステップ701で）、クラスタテーブルで読み取られる最初のクラスタL C が、オフセット値（バイト）をクラスタサイズ（バイト）で割ることによって計算される。次に、値C L U S T E R \_ O F F S E T が、クラスタサイズで割ったオフセット値の剩余として計算される（ステップ702）。次に、S E C T O R \_ I N \_ C L U S T E R が、C L U S T E R \_ O F F S E T をセクタサイズで割った値として計算される（ステップ703）。その後、値S E C T O R \_ O F F S E T が、C L U S T E R \_ O F F S E T をセクタサイズで割った剩余として計算される（ステップ704）。次に、読み取ろうとする論理セクタLOGICAL\_SECTOR が、次式によって計算される（ステップ705）：

$$\text{LOGICAL\_SECTOR} = (\text{CLUSTER\_TABLE} [\text{LC}] \times \text{クラスタ当たりのセクタ数}) + \text{SECTOR\_IN\_CLUSTER}$$

式中、C L U S T E R \_ T A B L E [L C ] は、クラスタテーブルの論理クラスタ L C によって参照されるデータクラスタを表す。次に、計算で求められたLOGICAL\_SECTOR 及びS E C T O R \_ O F F S E T 値を用いて、読み取ろうとするデータの場所が突き止められ、それらのデータが取り出される（ステップ706）。検索して取り出されたクラスタが最後に読み取ら

れるクラスタでなければ、論理クラスタLCが1だけインクリメントされ、新しいLOGICAL\_SECTOR R値を計算することによってこのルーチンが繰り返される（ステップ705）。検索して取り出されたクラスタが最後に読み取られるクラスタであった場合、ルーチンは終了する。

【0031】以上、本発明の高速DOSファイルシステムを実施形態により詳細に説明した。本願では、本発明をその特定の実施形態に関連して説明したが、特許請求の範囲の範囲に記載する本発明の広義の精神及び範囲から逸脱することなく、これらの実施形態について種々の修正態様及び変更態様を実施することが可能なことは明白であろう。したがって、本願の明細書及び図面は、限定的な意味ではなく、例示説明を目的とした意味で解釈すべきものである。

#### 【図面の簡単な説明】

【図1】搭載型自動車用ナビゲーションシステムのブロック図である。

【図2】DOSファイルアクセステーブル(FAT)の一例の説明図である。

\* 【図3】高速FAT(FFAT)の説明図(A)と、クラスタテーブルの説明図(B)と、ファイルのデータクラスタに対する参照を記憶したクラスタテーブルの説明図(C)である。

【図4】システムリセットまたはディレクトリの変更に応答してFFATの場所を突きとめるためのルーチンを図解したフローチャートである。

【図5】FFATを生成するためのルーチンを図解したフローチャートである。

10 【図6】ファイル用のクラスタテーブルを生成するためのルーチンを図解したフローチャートである。

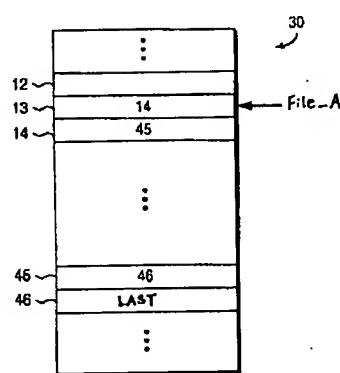
【図7】FFATを用いてファイル読み取るためのルーチンを図解したフローチャートである。

#### 【符号の説明】

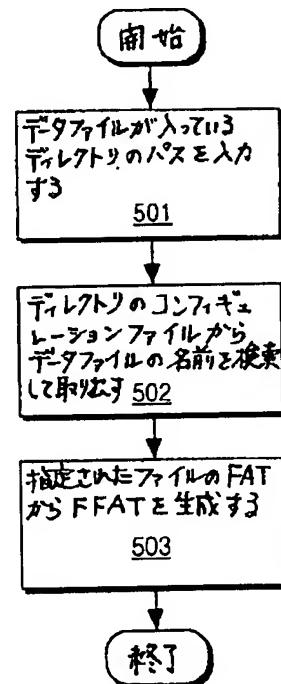
10 CPU、11 ROM、12 RAM、13 大容量記憶媒体、14 入力サブシステム、15 音声出力コントローラ、16 スピーカ、17 ディスプレイコントローラ、18 表示装置、19 角速度センサ、20 走行距離センサ、21 GPS。

\*20

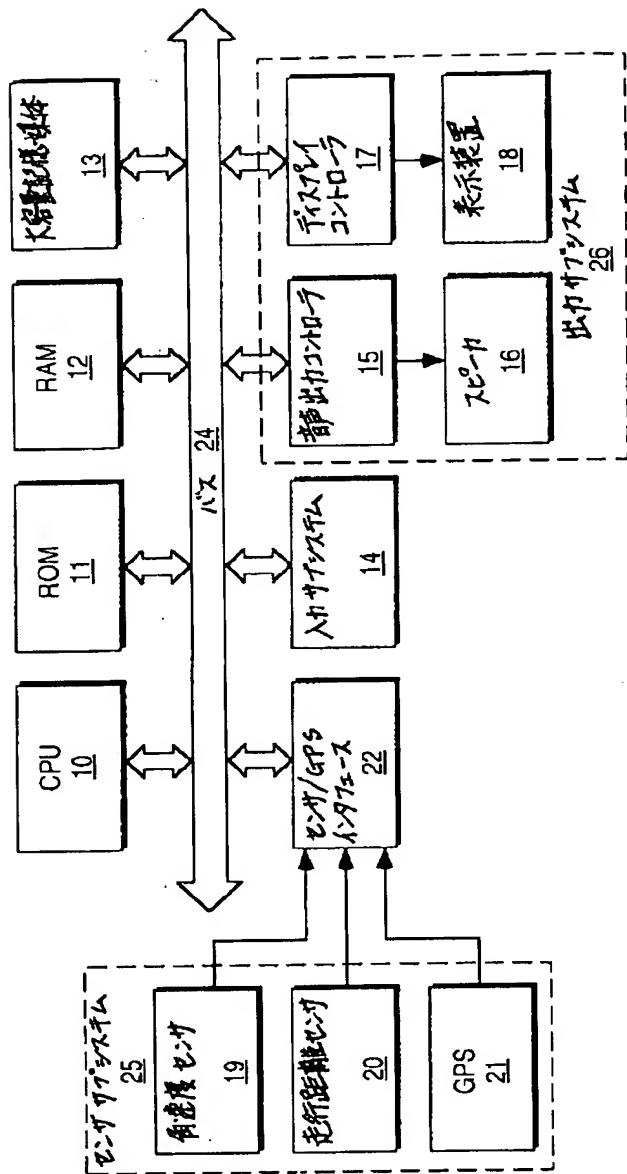
【図2】



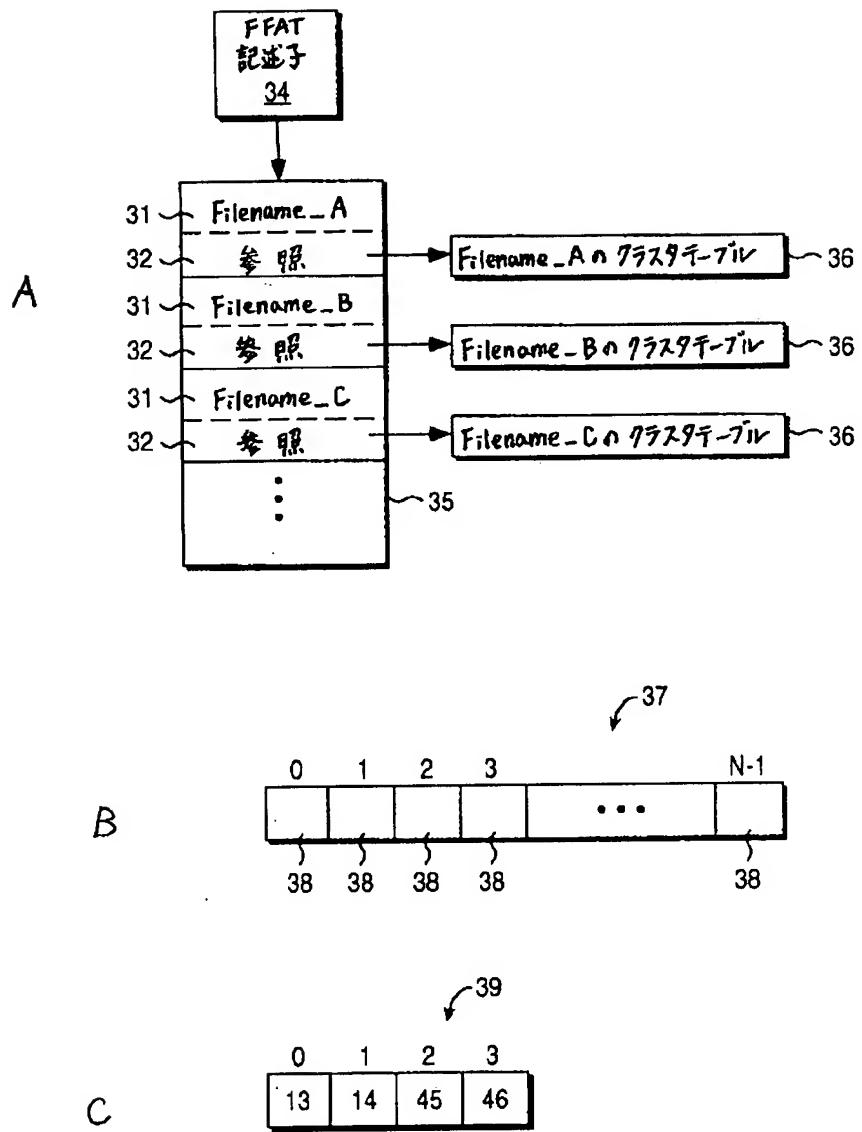
【図5】



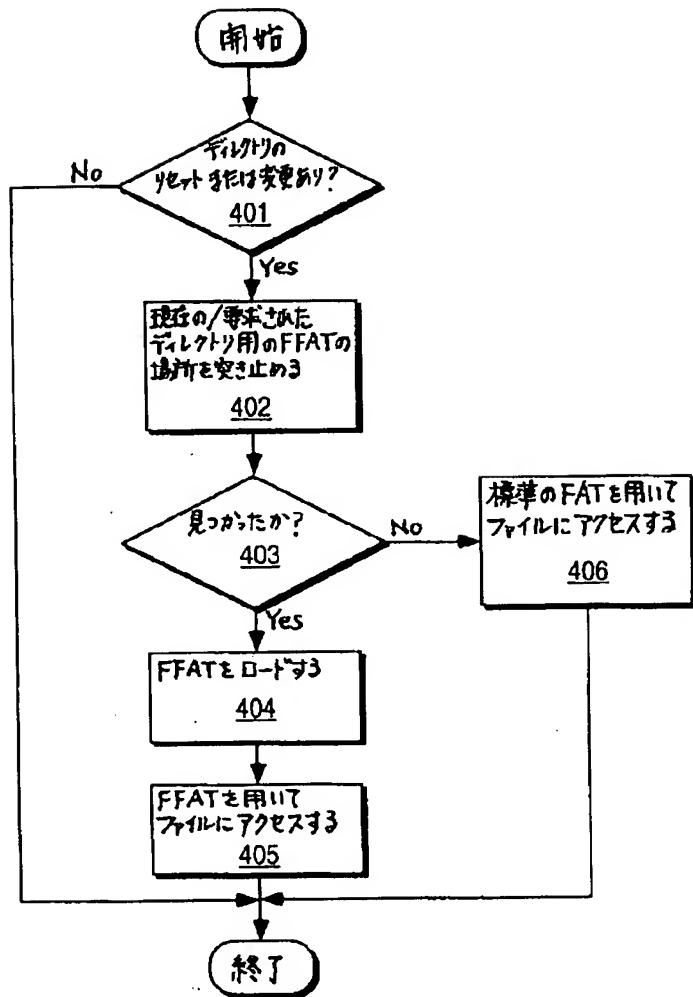
【図1】



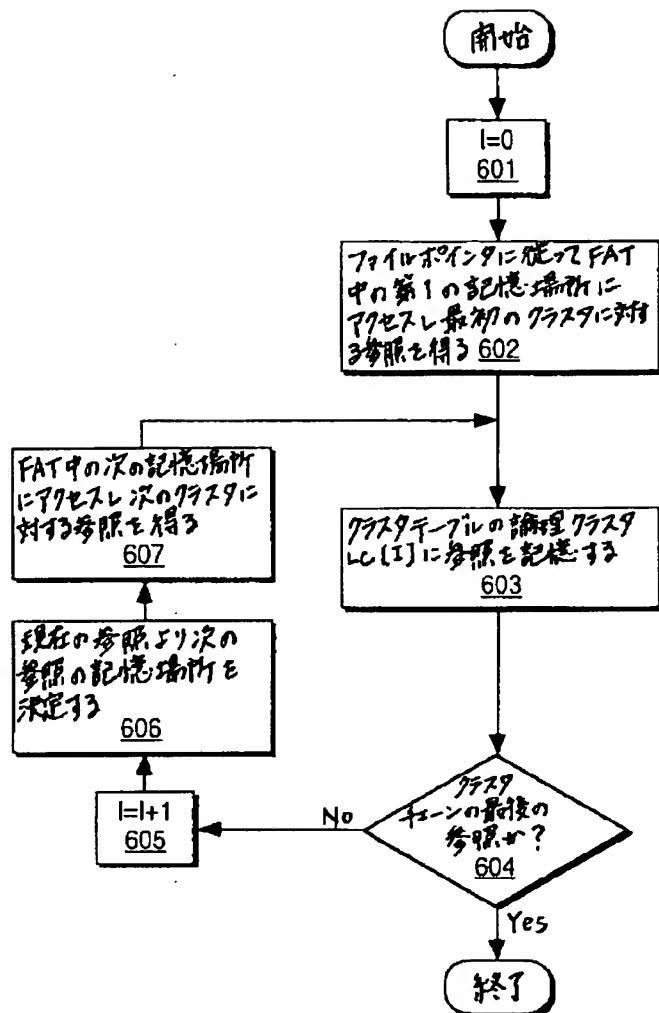
【図3】



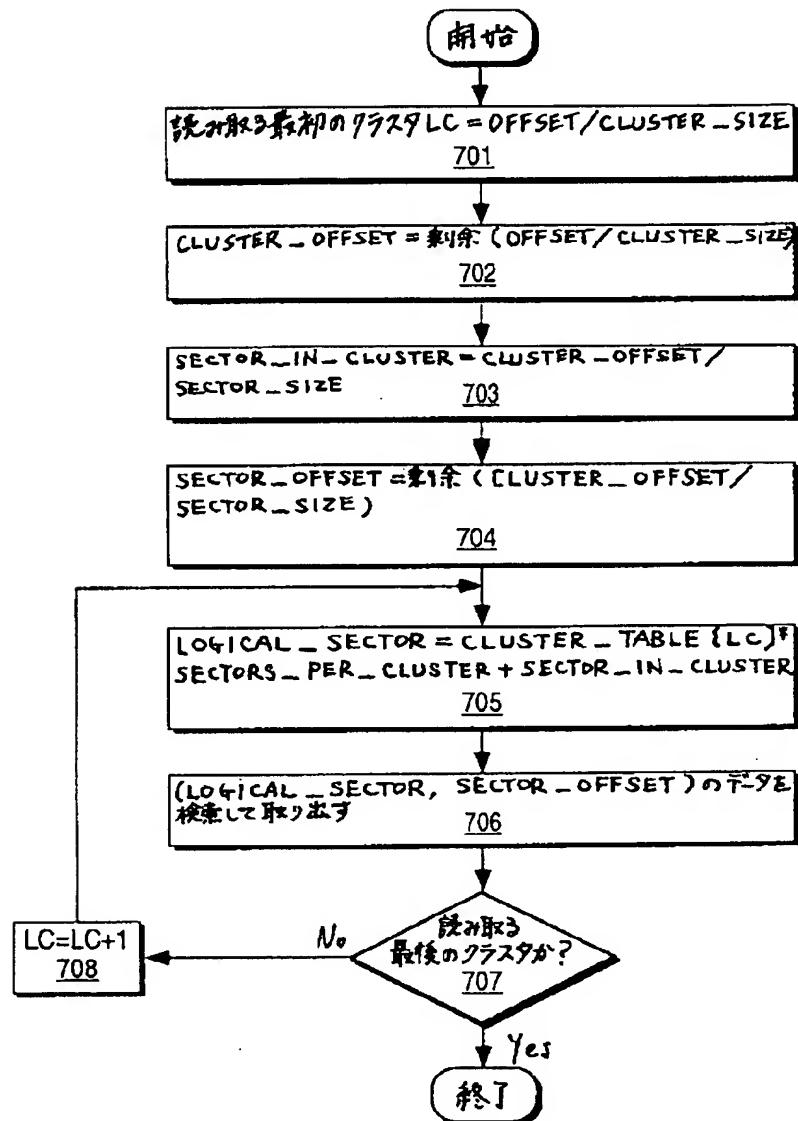
【図4】



【図6】



【図7】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-228400

(43)Date of publication of application : 25.08.1998

---

(51)Int.Cl. G06F 12/00

G01C 21/00

---

(21)Application number : 09-324429 (71)Applicant : ZEXEL KK

(22)Date of filing : 26.11.1997 (72)Inventor : LIAW JEFF JYH-MIN  
SAKAKIBARA TOSHIKAZU

---

(30)Priority

Priority number : 96 753483

Priority date : 26.11.1996

Priority country : US

---

(54) HIGH-SPEED FILE SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To enable high-speed data access by storing reference to subsets in successive storage locations of a 2nd table according to the logical order of the subsets in a file from a 1st table.

SOLUTION: A DOS(disk operating system) file system accesses the file by using a

cluster chain and stores it in a file access table as the 1st table of connection between the file and data clusters in the file. This file access table consists of a descriptor 34, a reference table 35, and a cluster table 36 of 2nd tables. In contrast with this file access table, reference 32 to subsets is stored in successive locations of the cluster table 36 according to the logical order of the subsets in the file.

---

LEGAL STATUS [Date of request for examination]  
[Date of sending the examiner's decision of rejection]  
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]  
[Date of final disposal for application]  
[Patent number]  
[Date of registration]  
[Number of appeal against examiner's decision of rejection]  
[Date of requesting appeal against examiner's decision of rejection]  
[Date of extinction of right]

\* NOTICES \*

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
  2. \*\*\*\* shows the word which can not be translated.
  3. In the drawings, any words are not translated.
- 

## CLAIMS

---

### [Claim(s)]

[Claim 1] The reference to :subset is included in the approach of enabling access to the file containing two or more subsets arranged by the logical order. The step which determines the logical order of the subset in a file from the 1st table arranged in the sequence that the reference differs from the logical order of the subset in said file; according to the logical order of the subset in said file, the reference to the subset The step memorized to the memory location which the 2nd table followed, and the approach possessing;.

[Claim 2] The step which memorizes the reference to said subset: The approach of; claim 1 publication that the location of referring to each in said 2nd table consists of a step which fears the reference to a subset an account in the memory location which the 2nd table followed according to the logical order of

the subset in a file as expresses the logic location of a subset where it corresponds in said file.

[Claim 3] The approach according to claim 1 which possesses further the step which accesses the memory location which the plurality in said 2nd table followed, and searches and takes out the cluster of said file.

[Claim 4] The approach according to claim 1 which possesses further the step which accesses said file, and searches and takes out at least one cluster of said file by accessing the memory location in said 2nd table at random.

[Claim 5] The approach according to claim 1 which possesses further the step which answers that said file is updated and updates said 2nd table automatically.

[Claim 6] The approach according to claim 1 said reference consists of a step which said 1st table memorizes in the form of a linked list, the step which determines the logical order of the subset in said file reads reference by the linked list, and specifies a subset, and specifies the logical order of the subset in a file.

[Claim 7] It is the approach of enabling it to access the file which contains two or more data subsets in a database using a file system. A file system contains the file access table (FAT) which has two or more memory locations. Each memory

location includes the reference to one of the data subsets, and the reference to another memory location in FAT. In the approach the reference to other memory locations in a file directs the following logic data subset in a file. The step which specifies the logical order of the data subset in said file based on said FAT; : By memorizing the reference to said data subset to the memory location which said table followed based on said logical order. The step which builds the table containing two or more memory locations, and the approach possessing;.

[Claim 8] The approach according to claim 7 which possesses further the step which accesses said file, and searches and takes out at least one subset of a file by accessing the memory location in said table at random.

[Claim 9] The approach according to claim 7 which possesses further the step which accesses the memory location which the plurality of said table followed, and searches and takes out the cluster of said file according to said logical order.

[Claim 10] The approach according to claim 7 which possesses further the step which answers renewal of said file and updates said table automatically.

[Claim 11] The approach according to claim 7 the memory location of FAT is a linked list.

[Claim 12] The approach according to claim 7 said data subset is a data cluster

respectively.

[Claim 13] The approach according to claim 12 of consisting of a step which the reference to said data subset is memorized by said FAT as a linked list, and the step which specifies the logical order of the data subset in said file reads a linked list, and specifies the logical order of the data subset in a data file.

[Claim 14] The approach according to claim 7 said file system is a DOS file system.

[Claim 15] It is an approach for accessing the data file which contains the cluster of two or more data in a database using a file system. A file system contains the file access table (FAT) which has two or more memory locations. Each memory location includes the reference to another memory location in FAT which directs the next cluster of the reference and the data file to one of the clusters of a data file. In an approach with the reference able for the discontinuous memory location in FAT to have to the cluster of a data file The step which specifies the logical order of the cluster in said data file from said FAT; : By memorizing the reference to said cluster to the memory location which the cluster table followed based on said logical order The step which builds a cluster table; the step which accesses said data file, and searches and takes out the cluster of a data file by

accessing serially to the memory location followed in said cluster table, and the approach possessing;.

[Claim 16] The approach according to claim 15 which possesses further the step which answers that said data file is updated and updates a cluster table automatically.

[Claim 17] The approach according to claim 15 by which the reference to said cluster is memorized by FAT as a linked list.

[Claim 18] The approach according to claim 17 the step which specifies the logical order of said cluster consists of a step which reads said linked list and specifies the logical order of the data cluster in said data file.

[Claim 19] It is the step which accesses the DOS file access table (FAT) which has the memory location of plurality in the approach of accessing the file which contains two or more data clusters in a database using a DOS file system. Each memory location includes the reference to one and FAT of a data cluster. The reference to a data cluster is memorized by FAT as a linked list. These reference The step which not necessarily specifies the logical order to which a data cluster appears in a file when all follow the step and the; aforementioned linked list which are not memorized by the memory location which FAT followed; a cluster

table including the reference to said data cluster By memorizing the reference to a data cluster to the memory location which this cluster table followed according to the logical order of the cluster in said file The step which builds the cluster table for a file; the step which accesses a file, and searches and takes out the cluster of a file by accessing the memory location which said cluster table followed, and the approach possessing;.

[Claim 20] The approach according to claim 19 said database is a map database of the street of a certain area.

[Claim 21] The approach according to claim 20 mounted in the vehicle loading system for supporting that the operator of a vehicle makes a course decision.

[Claim 22] In the equipment for supporting that the user of a vehicle determines the course of a vehicle : Provide the output subsystem for outputting course information to said user connected to said processor, and;. a processor, the database which has a data file containing two or more data subsets which have; logical order, and; -- the sensor subsystem connected to said processor, and; -- It is constituted. said processor -- : -- the data from said sensor subsystem and said database -- being based -- the current position of a vehicle -- calculating --; --; which makes the directions information on the current position output to said

output subsystem to a user -- it needs -- Said processor includes the reference to the :aforementioned data subset further. The reference The logical order of the data subset in said data file The method of; which memorizes the reference to said data subset to the memory location which the logical order of the data subset in a data file was determined, and the 2nd table followed according to the logical order of the data subset in the; aforementioned data file from the 1st table arranged in different sequence Equipment constituted.

[Claim 23] Equipment according to claim 22 said whose database is a map database of the street of a certain area.

[Claim 24] Equipment according to claim 23 with which said data file includes the information which identifies the subset of a street.

[Claim 25] Equipment according to claim 22 constituted so that the relative location of referring to each in said 2nd table may express further the relative location of a cluster where said data file in said logical order corresponds and said processor may memorize the reference to a data subset to the memory location which the 2nd table followed according to the logical order of the data subset in a data file.

[Claim 26] Equipment according to claim 22 constituted so that a processor may

access said file and may search and take out at least one of the files by accessing the memory location of said 2nd table at random further.

[Claim 27] Equipment according to claim 26 constituted so that said processor may access said file and may search and take out the cluster of a file according to said logical order by accessing further the memory location which the plurality of said 2nd table followed.

[Claim 28] Equipment according to claim 22 constituted so that said processor may answer further that said file is updated and may update said 2nd table automatically.

[0029] Equipment according to claim 2 with which said reference is memorized by said 1st table in the form of a linked list.

[Claim 30] Equipment given in a claim constituted so that a processor may read reference of said 1st table, and may carry out data-subset discernment and may specify the logical order of the data subset in a data file further.

[Claim 31] In the machine-readable program storage which mounted the program of the instruction which can be executed on the computer system for performing the approach step which enables access to the data file in the database containing two or more data subsets arranged by the logical order :

reference When arranged in the 1st table in different sequence from the logical order of the data subset in a data file, It determines from the 1st table including the reference of as opposed to a data subset for the logical order of the data subset in said data file. The machine-readable program storage which recorded the program which memorizes the reference to this data subset on the memory location which the 2nd table followed according to the logical order of the data subset in a data file.

[Claim 32] the time of memorizing the reference to said data subset -- : -- the machine-readable program storage according to claim 31 which memorizes the reference to a data subset to the memory location which the 2nd table followed according to this logical order of the data subset in this data file so that the relative location of a cluster where said data file which the relative location of referring to each in said 2nd table can boil and set by said logical order corresponds might be expressed.

[Claim 33] Furthermore, a machine-readable program storage including accessing said file, and searching and taking out the cluster of a file according to said logical order by accessing at random the memory location followed in said 2nd table, according to claim 31.

[Claim 34] Furthermore, a machine-readable program storage including the thing for which said file is updated and which carry out a thing response and updates said 2nd table automatically according to claim 31.

[Claim 35] The machine-readable program storage according to claim 31 with which said reference is memorized by said 1st table in the form of a linked list.

[Claim 36] The machine-readable program storage according to claim 32 which reads the reference in said 1st table, specifies a data subset, and specifies the logical order of the data subset in a data file in case the logical order of the data subset in said data file is determined.

[Claim 37] Processor; It is the 1st storage with which the database connected to said processor is memorized. The data file in which the database contains two or more data clusters is included. It is referred by the 1st table on which a data file has two or more memory locations. Each memory location includes the reference to at least one data cluster, and the reference to another memory location in the 1st table. The 1st storage which has the reference to the cluster of a data file in the discontinuous memory location in the 1st table; When it is the 2nd storage connected to said processor and a processor performs, To the processor : Make the logical order of the data cluster in said data file specify

from said 1st table, and it is based on the; aforementioned logical order. Said data file is accessed by making the 2nd table build and accessing at least one continuous memory location in the table of the; above 2nd at random by memorizing the reference to said cluster to the continuous memory location. The 2nd storage which memorized; instruction which makes at least one cluster of a data file search and take out, and the computer system possessing;.

[Claim 38] The computer system according to claim 37 which is the file access table (FAT) on which said 1st table is maintained by the DOS file system.

[Claim 39] The computer system according to claim 37 in which said instruction includes further the instruction which it answers [ instruction ] that said file is updated and makes a processor update said cluster table automatically when a processor performs.

[Claim 40] The computer system according to claim 37 the reference to said data cluster is remembered to be by said 1st table as a linked list.

[Claim 41] The computer system according to claim 37 as which said instruction in which the logical order of said data cluster is made to specify it as said processor makes this logical order of the data cluster in a data file specify by making a processor read a linked list further.

[Claim 42] The computer system according to claim 37 said whose database is a map database of the street of a certain area.

[Claim 43] The computer system according to claim 42 in which said data file includes the information which identifies the subset of said street.



## DETAILED DESCRIPTION

---

### [Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to a computer system. This invention relates to the high-speed file system for accessing the database of the navigation system for loading mold vehicles in more detail.

[0002]

[Description of the Prior Art] A loading mold navigation system can use deciding a course for helping, in case an automobilism person runs through unfamiliar land. It is equipment of the computer base where an example of such a system makes the destination choose from the street map database memorized by the operator in, makes the root to the selected destination calculate in, and leads an operator to the selected destination after that. Such a system has the various sensors for generally determining the location of a vehicle. In connection with a vehicle moving, this system processes data from a sensor and determines the location of the probable vehicle to map data. This system can display the indicator in which the location of a vehicle and a motion are shown on the map

which could display the street map of a surrounding area on the operator, and was displayed, in case a vehicle moves. Moreover, this system can also give an operator the course directions by voice and/or vision. Such a navigation system for loading mold automobiles is available from the ZEXEL U.S.A. corporation (Zexel USA Corporation) company of current and U.S. California Sunnyvale (Sunnyvale).

[0003] This kind of loading mold navigation system is controlled by the computer. Other computers and a computer [ in / similarly / the navigation system for loading mold vehicles ] have the operating system which is the software for controlling fundamental actuation of a system. An operating system names a file and the file system for managing a file, such as composing, memorizing and searching a file, is in it by \*\*. However, there is a problem on some designs about the file system of the navigation system for loading mold vehicles. Generally these problems relate to whenever [ rate / of a file system /, modifiability, and memory use ], and, a commercial-scene support.

[0004] For example, an Original-Equipment-Manufacturing contractor (OEM) may purchase a file system from external supply origin. However, in case use of the proprietary software supplied from the external source changes file system

software or decompiles it, it may have to receive application of some constraint imposed by the supplying agency. Such constraint becomes the hindrance at the time of debugging actuation, or bars the upgrade of the whole navigation system. Moreover, about the file system which can assert ownership, it is sometimes difficult to find an exchange tool in a commercial scene. Therefore, it may be desirable to use the file system which has an open architecture (namely, official announcement specification).

[0005] The well-known DOS (disk operating system) file system has an open architecture, therefore the DOS file system has become an existence attractive as a future file system for loading mold navigation systems. Moreover, a DOS file system needs only the comparatively small amount of memory as compared with the file system which can assert a part of ownership. However, a DOS file system tends to become late, when accessing a large file so that it may be detailed and may state later (it indicates in the term of "the gestalt of implementation of invention"). This delay when accessing a file may divert an operator's attention, therefore may cause risk. So, a standard DOS file system is not suitable to use it by the navigation system for loading mold vehicles generally.

[0006] Therefore, a high-speed data access is possible, whenever [ memory use ] is small, and the file system with which flexibility and a commercial-scene support are secured is needed. Furthermore, the high-speed file system which has an open architecture and was suitable for use by the navigation system for loading mold vehicles is also needed.

[0007]

[Means for Solving the Problem] This invention is the approach of enabling access to a file. A file contains in a file the subset of a large number put in order by the logical order. The 1st table includes the reference to a subset. The logical order of the subset in a file is determined from the 1st table. Reference is put in order in different sequence from the logical order of the subset in a file in the 1st table. Next, the reference to a subset is memorized by the memory location which the 2nd table followed according to the logical order of the subset in a file.

[0008] About the description of others of this invention, I will become clear from the detailed explanation about that of an accompanying drawing and a next operation gestalt.

[0009] The detailed explanation by the operation gestalt of this invention shown in the following accompanying drawings is a thing for instantiation explanation, it

does not have restrictive semantics to this invention, and the same reference mark has shown the same component among drawing.

[0010]

[Embodiment of the Invention] Hereafter, an operation gestalt explains the high-speed DOS file system of this invention to a detail. In the following explanation, in order [ that this invention is perfect ] to make it understand, many concrete detail matters are indicated. However, probably, for this contractor, it will be clear for this invention to be able to carry out without the publication of these concrete detail matters. In order to give explanation of this invention easy in the case of others, a block graphic form type shows a well-known configuration and equipment.

[0011] The technique for improving the effective data-access rate of a DOS file system is included in this invention so that it may explain to a detail after this. In 1 operation gestalt, the high-speed DOS file system of this invention is mounted in the navigation system for loading mold automobiles. However, various this inventions including architecture unrelated to the navigation system for loading mold vehicles can be carried out by different architecture. For example, this invention can be mounted in a personal computer.

[0012] Overall system-architecture drawing 1 illustrates the navigation system 1 for loading mold automobiles which mounted this invention with a block diagram. The navigation system 1 of illustration has a central processing unit (CPU) 10, a read only memory (ROM) 11, random access memory (RAM) 12, and a mass storage medium 13, and these are all connected to the bus 24. A bus 24 may consist of two or more buses each other connected by the bridge and/or the bus adapter in fact. Moreover, the output subsystem 26 is also connected to the bus 24. Although an output subsystem 26 receives a signal from the voice output controller 15 connected to the bus 24, it contains the audio loudspeaker 16. Moreover, an output subsystem 26 receives a signal including VDU 18 from the display controller 17 by whom this indicating equipment was connected to the bus 24. The input subsystem 14 is connected to the bus 24, and this input subsystem gives a means by which a user (for example, the operator or PAX of a vehicle) inputs an input into a navigation system 1. Moreover, the sensor subsystem 25 is connected to the bus 24 through the interface device 22. The sensor subsystem 25 has the receiver 21 of the angular-velocity sensor 19, the transit mileage sensor (for example, odometer) 20, and all earth accumbency systems (GPS), and these are respectively connected to the interface device 22.

An interface device 22 formats the data received from sensors 19 and 20 and the GPS receiver 21, and transmits them to a bus 24.

[0013] About each component shown in drawing 1, alternatives various by within the limits of this invention are possible. For example, a mass storage medium 13 is CD. Including a magnetic disk or optical storage disks, such as ROM (CD-ROM) (compact disk ROM), CD-R (recordable CD), and DVD (digital multi-purpose videodisk), if suitable, the nonvolatile storage of the gestalt of arbitration can be used. In 1 operation gestalt of this invention, a mass storage medium 13 is a PCMCLA (Personal Computer Memory Card International Association; personal computer memory card international association) hard disk drive. Furthermore, a mass storage medium 13, and ROM11 and RAM12 may express two or more physical units, respectively. Similarly, the Braun tube (CRT), a liquid crystal display (LCD), or the suitable vision output unit of the gestalt of other arbitration can be used for an indicating equipment 18. In 1 operation gestalt, an input subsystem 14 contains the mechanical carbon button or key which a user can input alphanumeric information into a navigation system 1 by it, or can choose the information displayed on the indicating equipment 18. With other operation gestalten, it can replace with a mechanical carbon button

and a pressure-sensitive carbon button or a sensible-heat carbon button, a touch sensitive screen, voice actuation input system, or another alternative means can be used. Furthermore, the specific sensor subsystem 25 shown in drawing 1 can be transposed to the detection equipment of other gestalten within the limits of this invention. For example, the GPS receiver 21 can be transposed to the high precision positioning system of other gestalten, can be replaced with an angular-velocity sensor, and can use a bearing detector (for example, compass) absolutely.

[0014] The navigation system 1 of the whole system of operation has the database of the data which carry out the discernment directions of the locations for an interest, such as street map data, a sightseeing spot, a restaurant, an airport, and a shopping mall, and other data. Street map data contain the data showing the scale from which versatility differs, and the scale of detail drawing. A database is memorized by the mass storage medium 13 in 1 operation gestalt. By this navigation system 1, an operator can calculate the root to that selected destination, after choosing the desired destination from the memorized database.

[0015] During transit of a vehicle, a display 18 is used for a system 1 and it displays the street map of a predetermined area to an operator. The directions

mark which shows the current position of the vehicle to the displayed map and a motion is included in this display. In case a vehicle runs, a system 1 processes the data from the angular-velocity sensor 19 and the transit mileage sensor 20, and the technique called the "dead-reckoning navigation" which asks for change of the relative position from the location calculated before is used. Next, a system 1 corrects location measurement for the location on count if needed as compared with a map database using a map matching algorithm. The positioning result obtained using dead-reckoning navigation and map matching depending on the case can also be corrected, amended or permuted using the data from the GPS receiver 21. In connection with a vehicle running, a system outputs course directions of the form of record voice or synthesized speech through a loudspeaker 16 to a user. Moreover, course directions can also be outputted to a user with a display 18.

[0016] A navigation system 1 functions based on an operating system. An operating system has a file system for performing naming of a file, organization, storage, and a function like retrieval. This invention includes the technique for improving actuation of a file system so that it may explain below.

[0017] In 1 operation gestalt, this invention answers and functions on performing

the sequence of the instruction with which the CPU10 was memorized by memory (the combination of ROM11, RAM12, mass storage media 13, or these equipments is sufficient as memory) by the navigation system 1. That is, CPU10 performs the step of this operation gestalt explained below by activation of the sequence of the instruction memorized by memory. For example, to load to RAM12 from ROM11 or \*\*\*\*\* storage like a mass storage medium 13, and to perform can also be ordered. In other operation gestalten, it is also possible by replacing with a software instruction or using a hard-wired circuit combining these software instructions to carry out this operation gestalt. thus, this invention is not limited to a certain specific combination of hardware circuitry and software, and is limited to what kind of the specific source of the instruction executed by the navigation system 1 -- also coming out -- there is nothing.

[0018] The technique for improving the effective speed of a DOS file system is included in file access table this invention. Since the data-access rate of a DOS file system is comparatively slow, generally it is not suitable as an object for automobiles for navigation systems. Here, the reason nil why this data-access rate is comparatively slow is explained.

[0019] Generally data are memorized on a storage in the form of a data cluster.

A cluster means the physical sector which some of storages followed. The data cluster of each file has a logical order in the file. Generally a DOS file system accesses a file using a cluster chain. A cluster chain is the linked list of the data cluster in a specific file. These cluster chains are memorized by the file access table (FAT) which is association between the data clusters of a file and its file.

[0020] FAT30 is illustrated by drawing 2. FAT30 has much memory locations and the reference to the specific data cluster of a file is memorized by each of these memory locations. Referring to each is also reference to the following data cluster by the logical order in a file. It is shown by the entry point the start of the cluster chain in each file, and this entry point is the memory location 12 in FAT30. It follows. The cluster chains of File\_A are (13, 14, 45, 46). The cluster of the last of the cluster chain of a file is identified by the suitable identifier memorized by the memory location of the last of a cluster chain like the identifier "LAST" memorized by the memory location 46 of FAT30.

[0021] A file is accessed by performing forward search serially through the cluster chain connected with FAT to the file. The file pointer which directs the data cluster by which current reference is carried out can do \*\*\*\*\* moved to the front along with a cluster chain. When related data are behind a file pointer, a file

pointer can be set at the beginning of a cluster chain, and retrieval can be restarted. When a file is small, time amount required to perform this retrieval has that it is also so small that it can ignore. However, since a lot of data are generally contained, a database file requires time amount with comparatively long retrieval of a cluster chain in many cases. Therefore, FAT relevant to a DOS file system may be suitable for accessing the comparatively small file which has a short cluster chain. However, FAT is completely unsuitable, although a large data file which is usually looked at by the database of the navigation system for loading mold vehicles is accessed.

[0022] FAT "a \*\* top" of a file system (on a function) High-speed file access table  
this invention improves the data-access rate of a DOS file system effectively by generating the placed additional structure. In this application, this structure is called "a high speed FAT (FFAT)." An example of FFAT \*\* is shown in drawing 3 A. This structure is memorized by the new table which the cluster chain for every file calls a "cluster table" by this application. A cluster table has the memory location of a large number in which the reference to the real cluster in a file is written, i.e., a logical cluster. The reference to a cluster is memorized by the memory location which the cluster table followed according to the logical order of

the cluster in a file as contrasted with FAT of the criterion of a DOS file system. Consequently, a file can be accessed with a random method not using the serial method which used FAT but using the cluster table. The cluster table of \*\* is generated by traversing or scanning FAT relevant to the file about a predetermined file so that it may explain below.

[0023] Next, in drawing 3 A, FFAT consists of cluster tables 36 about some files [ like ] which are a descriptor 34, the reference table 35 and "Filename\_A", "Filename\_B", and "Filename\_C." The reference to the file name of each file and the cluster table for every file is written in the reference table 35. A reference mark "31" shows the file name in the reference table 35, and the reference mark "32" has shown the reference to a cluster table. A descriptor 34 describes the number of files in FFAT, and the size of the reference table 35. Thus, each file in FFAT33 has an entry on the reference table 35. An entry can be identified by the file name and the cluster table of a file can trace a location by the related reference. Thus, the cluster table about a predetermined file can be accessed by the file name.

[0024] The cluster table is shown more in the detail at drawing 3 B. The cluster table 37 has the memory location 38 of N individual called a "logical cluster."

Each logical cluster 38 in the cluster table 37 is identified with 0, 1, 2, ..., the index value that is within the limits of N-1. The index value of each logical cluster expresses the sequence that the data cluster referred to by the logical cluster appears in a logical order in a data file. For example, the cluster table 39 relevant to File\_A which is the file of an example explained by drawing 2 is shown in drawing 3 C. As stated above, the cluster chains attached File\_A are (13, 14, 45, 46). Thus, the reference to the data clusters 13, 14, 45, and 46 is included in these logical clusters including four logical clusters from which the cluster table 39 of File A is discriminated as logical clusters 0, 1, 2, and 3 (an index is carried out), respectively.

[0025] In 1 operation gestalt, FFAT is treated as an optional feature used by the DOS file system, only when available. When FFAT is not available, a file system is accessed at a file, only using standard FAT. With this operation gestalt, it tries for a file system to load FFAT for default directories, whenever a system 1 is reset. It tries similarly, for a file system to load the new FFAT table for directories, whenever a change of a directory is made. Modification of a directory is needed for accessing the data showing the area for example, under current access, and a different area. If loading of FFAT is performed well, it will change to table

access of FAT of the criterion about the file specified by the configuration file, and access of FFAT will be performed. When a FFAT table is not loaded well or the file is not specified by the configuration file, access of the usual FAT will be performed. Therefore, even if a navigation system 1 has no FAT table, it can operate. However, the rate becomes slower than the case where a FFAT table is used.

[0026] Next, this function is explained with reference to drawing 4 which shows the routine for tracing the location of FFAT. Reset of a navigation system 1 tends to be answered or (step 401) and a navigation system 1 tend to trace a current directory or the location of FFAT for directories demanded to modification of a directory and coincidence (step 402). If the FFAT is found (step 403), a system loads the corresponding FFAT (step 404). Next, a system accesses a suitable file using the FFAT (step 405). When FFAT is not found, a system is accessed at a suitable file using standard FAT (step 406).

[0027] FFAT must also be updated whenever the data file referred to by FFAT is updated. The routine for updating FFAT is shown in drawing 5. In 1 operation gestalt, the procedure of drawing 5 is carried out in the exclusive software utility for this purpose. First, the pass of the directory containing a data file is inputted

(step 501). Next, the identifier of the updated data file searches and is taken out from the configuration file of a directory (step 502). Next, updated FFAT is generated from FAT by scanning the cluster chain specified by FAT (step 503). The step (step 503) which updates FFAT includes the actuation which updates the cluster table of a file influenced by it.

[0028] Drawing 6 shows the routine for generating or updating the cluster table of a file. The index variable I is first set to zero (step 601), a system 1 accesses the 1st memory location of FAT based on a file pointer, and the reference to the 1st cluster is obtained (step 602). Next, this reference is memorized by logical cluster LC [I] (logical cluster which has the index value of I) of a cluster table. A routine will be ended if reference is the cluster of the last of a cluster chain (step 604). If it is not the last cluster, the increment of the index variable I will be carried out only for 1 (step 605). After incrementing the index value I, it decides on the memory location of reference of the degree in a cluster chain from current reference (step 606). Next, the following memory location in FAT is accessed and the reference to the following cluster in the cluster chain is obtained (step 607). Then, a routine is repeated from step 603.

[0029] a data file reads the logical cluster of the cluster table of a file serially -- it

is accessed. A data access is started by the read demand to a file system. A file name, offset, and a count are included in a read demand. A file begins offset and it is measured from a predetermined point in a file like the current position or an end of file. It is :READ ([FILE\_NAME], [OFFSET], [COUNT]) in which a read demand has the following formats in 1 operation gestalt by a count specifying the number of data bytes read therefore. Offset can be given, and it is calculated when the initiation logical cluster of a cluster table breaks offset by cluster size. Thus, the initiation logical cluster of a cluster table is serially accessed by not a method but random. Next, the reference to a live-data cluster can be searched and taken out from the logical cluster of a cluster table. The sector number relevant to offset is calculable by multiplying the real cluster number obtained from the cluster table by the number of sectors per cluster. By supplying a logical sector number and the number of sectors to a device driver, the map of the logical sector can be carried out to a physical sector, and data can be searched and picked out from storage. The logical cluster in a cluster table is read by the order which appears in a cluster table. Thus, a file is accessed not as a sequential access file but as a random access file.

[0030] Drawing 7 shows the example of the routine of a data reading \*\*\*\* sake

by the approach described above. It is calculated when introduction and the first cluster LC read on a cluster table (at step 701) break an offset value (cutting tool) by cluster size (cutting tool). Next, value CLUSTER\_OFFSET is calculated as a remainder of the offset value broken by cluster size (step 702). Next, SECTOR\_IN\_CLUSTER is calculated as a value which broke CLUSTER\_OFFSET by sector size (step 703). Then, value SECTOR\_OFFSET is calculated as a remainder which broke CLUSTER\_OFFSET by sector size (step 704). Next, :LOGICAL\_SECTOR by which logical sector LOGICAL\_SECTOR which it is going to read is calculated by the degree type (step 705) The inside of a =(number of sectors per CLUSTER\_TABLE[LC] x cluster)+SECTOR\_IN\_CLUSTER type and CLUSTER\_TABLE [LC] are the logical cluster of a cluster table. LC The data cluster referred to is expressed. Next, the location of the data which it is going to read is traced using LOGICAL\_SECTOR and the SECTOR\_OFFSET value required in count, and those data are taken out (step 706). If the cluster searched and taken out is not a cluster read at the end, the increment of the logical cluster LC will be carried out only for 1, and this routine will be repeated by calculating a new LOGICAL\_SECTOR value (step 705). A routine is ended when the cluster

searched and taken out is a cluster read at the end.

[0031] In the above, the operation gestalt explained the high-speed DOS file system of this invention to the detail. In this application, although this invention was explained in relation to the specific operation gestalt, probably, it will be clear that it is possible to carry out various correction modes and modification modes about these operation gestalten, without deviating from the pneuma and the range of a wide sense of this invention indicated in the range of a claim. Therefore, the specification and drawing of this application should be interpreted not in restrictive semantics but in the semantics aiming at instantiation explanation.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of the navigation system for loading mold automobiles.

[Drawing 2] It is the explanatory view of an example of a DOS file access table (FAT).

[Drawing 3] They are the explanatory view (A) of a high speed FAT (FFAT), the explanatory view (B) of a cluster table, and the explanatory view (C) of the cluster table which memorized the reference to the data cluster of a file.

[Drawing 4] It is a flow chart illustrating the routine for answering modification of a system reset or a directory and tracing the location of FFAT.

[Drawing 5] It is a flow chart illustrating the routine for generating FFAT.

[Drawing 6] It is a flow chart illustrating the routine for generating the cluster table for a file.

[Drawing 7] It is the flow chart which illustrated the routine of a file reading \*\*\*\* sake using FFAT.

[Description of Notations]

10 CPU, 11 ROM, 12 RAM, 13 A mass storage medium, 14 input subsystems, 15 A voice output controller, 16 A loudspeaker, 17 A display controller, 18 A display, 19 An angular-velocity sensor, 20 A mileage sensor, 21 GPS.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: \_\_\_\_\_**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**